

Region-Based Fractal Image Compression

Hannes Hartenstein, *Associate Member, IEEE*, Matthias Ruhl, and Dietmar Saupe

Abstract—A fractal coder partitions an image into blocks that are coded via self-references to other parts of the image itself. In this paper we present a fractal coder that derives highly image-adaptive partitions and corresponding fractal codes in a time-efficient manner using a region-merging approach. The proposed merging strategy leads to improved rate-distortion performance compared to previously reported pure fractal coders, and it is faster than other state-of-the-art fractal coding methods.

Index Terms—Fractal compression, image compression, image partitions, segmentation-based coding.

I. INTRODUCTION

IN fractal image compression an image is modeled as the unique fixed point of a contractive operator on the space of images. This type of image representation was first proposed by Barnsley and Sloan [1], [2] and Jacquin [3], [4] who devised the first practical fractal coder. Fractal coding has since been a topic of active research because it has opened up a refreshing new view to image compression. It leads to visually pleasing results at high compression ratios, and it provides resolution independent image descriptions.

In fractal compression the image to be encoded is partitioned into image blocks called ranges. Each range is “coded” by a reference to some other part of the image and by some transformation parameters. These parameters describe how the referenced image part has to be adjusted with respect to contrast and brightness in order to give a good approximation to the range to be encoded. When one uses fixed length codes for the transformation parameters, the size of a fractal code is, up to the code for the partition, proportional to the number of blocks of the partition. Therefore, in order to obtain high compression ratios, only a small number of blocks are allowed. Thus, the key point in fractal compression is to partition the image into a small number of blocks that are *similar* to other image parts under certain transformations. Since it has been shown that the problem of determining the best piecewise constant approximation for a given number of ranges represents an NP-hard problem [5], it follows that finding optimal partitions for fractal coding purposes is at least as hard. Many partitioning methods have been proposed for fractal image compression; among the most widely known and

most successful are Fisher’s quadtree scheme [6] and Fisher’s and Menlove’s horizontal–vertical partitions [7]. With these approaches, the partition is hierarchical and built in a top–down fashion.

In this paper we investigate highly image-adaptive partitions in order to improve the rate-distortion performance of fractal coding. The proposed fractal coder can be seen as a combination of segmentation-based image coding and fractal compression. The partitions are derived in a bottom-up approach using region merging. Compared to hierarchical tree-structured partitions a higher rate is required for encoding the irregular partitions. However, we will show that this investment pays off in terms of an improved rate-distortion performance. With our region-based fractal coder (RBFC) we build upon the approach of Thomas and Deravi [8]. The image is first uniformly partitioned, and then neighboring range pairs are successively merged reducing the total number of ranges one by one. Because of the large number of choices during the merging process, a heuristic strategy has to be applied. We have developed such a strategy for the region merging that performs well. Moreover, we have used an efficient coding scheme for the resulting partitions. The region merging strategy and the efficient partition coding have led to a much improved rate-distortion performance compared to the results reported in [8], e.g., a gain of about 5 dB PSNR is obtained for the 512×512 Lenna image at a compression ratio of 40 : 1.

A specific advantage of our RBFC approach is the significantly reduced encoding time compared to the other state-of-the-art fractal coders. The encoding time efficiency is due to a particular domain search heuristics and sophisticated algorithmic techniques that will be described in this paper.

The proposed fractal coding method represents a *pure* fractal coder in the sense that it does not employ a hybrid approach that combines fractal and transform coding. Nevertheless, our RBFC in combination with an additional full domain search gives rate vs distortion results that are comparable to those of the fractal-wavelet coder of Davis [9] and the fractal-DCT coder of Barthel [10]. With the proposed scheme we believe to exploit image-inherent self-similarities to the fullest; however, encoding times as well as rate-distortion performance are still not competitive with state-of-the-art wavelet coders. Preliminary reports of our work on region-based fractal coding were presented in [11]–[13].

This paper is organized as follows: in Section II we present a precise framework of fractal image representation and the fractal coding problem. While the basic fractal coding principles can easily be understood from a conceptual point of view, we feel that a solid mathematical formulation is indispensable for the sake of clarity and rigor. A review of previously used partitioning methods for fractal coding is given in Section III. Our

Manuscript received May 4, 1999; revised October 24, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nasir Memon.

H. Hartenstein is with the Computer & Communications Research Laboratories, NEC Europe Ltd., 69115 Heidelberg, Germany (e-mail: Hannes.Hartenstein@crrle.nec.de).

M. Ruhl is with the Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: ruhl@mit.edu).

D. Saupe is with the Institut für Informatik, Universität Leipzig, Leipzig, Germany (e-mail: saupe@informatik.uni-leipzig.de).

Publisher Item Identifier S 1057-7149(00)05319-7.

region-based fractal coder is presented in Section IV, and experimental results as well as comparisons to other fractal coders are discussed in Section V. In Section VI we discuss additional studies, namely a rate-distortion optimization approach and a probabilistic region merging version. Our conclusions are summarized in Section VII.

II. FUNDAMENTALS

A. Fractal Image Modeling

We start with a mathematical framework of fractal modeling of images. We will use the framework given by an operator as studied by Read [14] and Bajraktarevic [15] and will proceed from a rather general formulation to more and more restricted versions. The Read–Bajraktarevic operators have been previously used for generating fractal curves and surfaces by Dubuc [16] and Massopust [17], and for fractal compression by Bedford *et al.* [18].

Let X stand for the image support $[0, 1) \times [0, 1)$, and let the space of images $L^\infty(X, \mathbb{R})$ consist of all bounded functions $f: X \rightarrow \mathbb{R}$. The space $L^\infty(X, \mathbb{R})$ together with the metric $d_\infty(f_1, f_2) = \|f_1 - f_2\|_\infty = \sup_{x \in X} |f_1(x) - f_2(x)|$ represents a complete metric space. An operator $F: L^\infty(X, \mathbb{R}) \rightarrow L^\infty(X, \mathbb{R})$ is called contractive with respect to the d_∞ -metric if there exists a constant c with $0 \leq c < 1$ such that $\|F(f_1) - F(f_2)\|_\infty \leq c \cdot \|f_1 - f_2\|_\infty$ for all $f_1, f_2 \in L^\infty(X, \mathbb{R})$. For a contractive operator on a complete metric space the Banach fixed point principle guarantees the existence of a unique fixed point (cf. e.g., [19]). Thus, a contractive operator F on the space of images uniquely describes an image $\Omega_F \in L^\infty(X, \mathbb{R})$, namely the one that is characterized by the equation $F(\Omega_F) = \Omega_F$. Given a contractive operator $F: L^\infty(X, \mathbb{R}) \rightarrow L^\infty(X, \mathbb{R})$, the corresponding fixed point can be obtained by applying the operator F iteratively to an arbitrary image, i.e., the fixed point Ω_F is given by $\lim_{k \rightarrow \infty} F^k(f)$, $\forall f \in L^\infty(X, \mathbb{R})$. Since Ω_F represents an attractive fixed point, it is also called *attractor* of F .

In fractal image compression one is interested in (contractive) operators 1) whose attractors are good in approximating real world images and 2) that can be coded compactly. The class of block-based Read–Bajraktarevich operators have proven to meet these demands.

A Read–Bajraktarevich operator (RB-operator) $F: L^\infty(X, \mathbb{R}) \rightarrow L^\infty(X, \mathbb{R})$ is based on a self-mapping $g: X \rightarrow X$ that we call the *geometric mapping*, and on a function $v: X \times \mathbb{R} \rightarrow \mathbb{R}$, called the *value transformation*, and is given by

$$F(f)(x) = v(x, f(g(x))).$$

Thus, the value $F(f)(x)$ is determined using a place-dependent transformation of the grey value and some kind of self-reference contributed by the geometric mapping g .

A *block-based* RB-operator is based on a partition \mathcal{P} of the image support. A partition \mathcal{P} is a finite set of subsets of X such that $\bigcup_{P \in \mathcal{P}} P = X$, $\tilde{P} \cap P = \emptyset$ for $\tilde{P} \neq P$, and all $P \in \mathcal{P}$ are (path-)connected. With a block-based RB-operator for a

partition \mathcal{P} , the geometric mapping and the value transformation are of the form

$$g(x) = \sum_{P \in \mathcal{P}} g_P(x) \mathbf{1}_P(x), \quad v(x, y) = \sum_{P \in \mathcal{P}} v_P(y) \mathbf{1}_P(x)$$

where all g_P and v_P are affine mappings.¹ Here, $\mathbf{1}_P$ denotes the characteristic function of P . For simplicity, we will interpret the image support X as a torus, and the affine mappings g_P are then interpreted *modulo* 1. Each element P of the partition \mathcal{P} is called a *range*, whereas $g(P) = \{g(x) | x \in P\}$ is called a *domain*. Obviously, these designations come from studying the inverses of the functions g_P , $P \in \mathcal{P}$. In fractal compression literature one usually also calls $f \upharpoonright P$ a range and $f \upharpoonright g(P)$ a domain.

In order to deal with digital images in the above framework, further restrictions apply to the partition and the value transformation. Digital images are modeled as functions that are constant on each element of the uniform base partition \mathcal{U}_n

$$\mathcal{U}_n = \left\{ \left[\frac{k_v}{n}, \frac{k_v+1}{n} \right) \times \left[\frac{k_h}{n}, \frac{k_h+1}{n} \right), \right. \\ \left. 0 \leq k_v, k_h < n \right\}$$

for a suitable $n \in \mathbb{N}$. We call $U \in \mathcal{U}_n$ as well as $f \upharpoonright U$, $U \in \mathcal{U}_n$, a *pixel* and n the resolution of the image. Under the following conditions on the RB-operators one can avoid computations on a “sub-pixel” scale:

- partition \mathcal{P} is adapted to the resolution n , i.e., each $P \in \mathcal{P}$ is a (edge-connected) union of elements of \mathcal{U}_n . Thus, the base partition \mathcal{U}_n is a refinement of the partition \mathcal{P} ;
- geometric mapping g is adapted to resolution n , i.e., g maps each pixel to a set of pixels. Precisely, since we are dealing with half-open intervals, we require that for $U \in \mathcal{U}_n$ the closure $\overline{g(U)}$ equals the closure \overline{V} for some subset $V \subset \mathcal{U}_n$.

Assume that one is given a contractive block-based RB-operator $F = v(\cdot, g(\cdot))$ for a partition \mathcal{P} such that \mathcal{P} and g are adapted to resolution n . Let A_n denote the averaging operator that projects an arbitrary (measurable²) image onto the space of images of resolution n , i.e., the value of a pixel $f \upharpoonright U$ is obtained by integrating the image function over the pixel U . Then, the operator $F_n := A_n \circ F$ is also contractive and maps an image of resolution n to an image of the same resolution. Furthermore, the corresponding fixed point Ω_{F_n} satisfies

$$A_n(\Omega_F) = \Omega_{F_n}. \quad (1)$$

In other words, the fixed point of the operator F_n represents the “projection to resolution n ” of the real attractor, i.e., of the attractor of F . Equation (1) can be proven by taking into account the linearity of the integral operator and the special forms of functions v and g [21]. Also note that, when computing $F_n(f)$

¹The value transformations v_P can be generalized to place-dependent functions $v_P = v_P(x, y)$, see, e.g., [20].

²Since the averaging requires integration, one now has to be concerned with measurability issues which we will neglect here.

for $U \in \mathcal{U}_n$, one can first compute the average value of f on $g(U)$, and then apply the value transformation to this average value.

The above discussion shows that when one is given a block-based RB-operator with partition and geometric mapping adapted to resolution n , the corresponding fixed point can be easily decoded at resolutions $2^k n$, $k \in \mathbb{N}$. Thus, when an image is coded as the fixed point of such an operator F , it is represented in a *resolution independent* fashion. This is one of the special features of fractal image compression. The relationship between the finite resolution attractors and the real attractor has been first presented by Baharav *et al.* [22].

Now, let us focus on the specification of the geometric mapping and the value transformation for the case when one is given a block-based RB-operator $F = v(\cdot, g(\cdot))$ for a partition \mathcal{P} such that \mathcal{P} and g are adapted to resolution n . Since g is adapted, each affine mapping g_P , $P \in \mathcal{P}$, has to be of the form

$$g_P(x_1, x_2) = \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix} \mathcal{O} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

where

- a_1, a_2 positive integers;
- \mathcal{O} orthogonal transform that consists of a rotation of a multiple of 90° optionally followed by a reflection at the horizontal axis;

(b_1, b_2) is in the set $\{((k_v/n), (k_h/n)) \mid 0 \leq k_v, k_h < n\}$.³ In practice, one usually sets $a_1 = a_2 = 2$, as it has been done in all of our experiments. \mathcal{O} stands for one of 8 orthogonal transforms, the so-called *isometry options*. Usually, one restricts the set of feasible translational vectors (b_1, b_2) . In fact, instead of specifying a translational vector one gives the absolute coordinates—the so-called *domain address*—to which some reference point of the range, in our case the upper far left pixel of a range, is mapped. We use a grid for the reference point coordinates of the type

$$\left\{ \left(\frac{2^j k_v}{n}, \frac{2^j k_h}{n} \right) \mid 0 \leq k_v, k_h < \left\lfloor \frac{n}{2^j} \right\rfloor \right\}$$

with $1 \leq j \leq \lfloor \log_2 n \rfloor$, as set of feasible domain addresses. The corresponding set of domains for a given range is called the *domain pool* for the range; for $j = 1$ we refer to it by the name *full domain pool*. Note that for ranges differing in size and shape the domain pools for the ranges differ accordingly.

The parameters $s, o \in \mathbb{R}$ of an affine value transformation $v_P(y) = sy + o$ are called *scaling factor* and *offset*. They are also referred to as *luminance parameters*.

To summarize, the information needed to specify an RB-operator with the above restrictions comprises the following.

- Partition \mathcal{P} .
- Translational vector or the corresponding domain address for each g_P , $P \in \mathcal{P}$. When isometries are used, the specification for one of the eight choices is required.
- Scaling factor s_P and an offset o_P for each v_P , $P \in \mathcal{P}$.

This specification of an RB-operator is also called a *fractal code*.

³The reader is reminded of our convention of *wrap arounds*.

B. Inverse Problem

So far, we have discussed fractal image modeling. It remains to be shown how a fractal code can be constructed such that the corresponding attractor approximates a given image. Since one is interested in a compact representation, one is faced with the optimization problem of finding that RB-operator whose attractor gives the best approximation for a given bit rate necessary to specify the corresponding fractal code. Instead of aiming to find the truly optimal fractal code for a given bit rate, one is usually confronted with the following problem. Given a class of RB-operators—for example, the class of RB-operators based on a fixed partition \mathcal{P} , on a fixed set of domain addresses, and on a fixed (finite) set of feasible luminance parameters—one has to find in this class the operator whose attractor is closest to the input image. In other words, the rate is constrained by using a special subclass of RB-operators, and the quality of approximation has to be maximized. This optimization problem, called the *inverse problem of fractal compression*, is NP-hard [23]. Therefore, it is computationally infeasible to compute the optimal RB-operator and a heuristic method is applied instead, called collage coding [24].

Here, one searches in a feasible subclass of contractive (w.r.t. the d_∞ -metric) RB-operators for the one whose *collage* $F_n(f)$ minimizes the *collage error* $\|f - F_n(f)\|_2^2$. In detail this optimization works as follows. Assume that one is given an input image f of resolution n , a partition \mathcal{P} adapted to this resolution, and a quantizer for the luminance parameters. Then, in order to minimize $\|f - F_n(f)\|_2^2$ over the set of feasible RB-operators, one searches for the optimal geometric mapping and value transformation for each range independently of the other ranges. In order to compare a range $f \upharpoonright P$ to all possible domains in the domain pool of $f \upharpoonright P$, it is helpful to interpret the range as an element of \mathbb{R}^m , where m is the number of pixels of the range. Since each pixel of the range is compared to a 2×2 pixel block of a domain, one should first perform an averaging of the 2×2 blocks of the domain to obtain a so-called *codebook block*. A codebook block then can also be interpreted as an element of \mathbb{R}^m . The set of codebook blocks corresponding to a domain pool is called a *codebook*. Let R denote a range vector and let D stand for a codebook block vector. Then, the optimal luminance parameter \tilde{s}, \tilde{o} are obtained via

$$(\tilde{s}, \tilde{o}) = \arg \min_{s, o \in \mathbb{R}} \|R - (sD + o\mathbf{1})\|_2^2. \quad (2)$$

Here, $\mathbf{1}$ denotes a vector of dimension m with a 1 in each component. The optimal luminance parameters as given in (2) are quantized to get coefficients \tilde{s}_q, \tilde{o}_q ; the quantizer also has to guarantee that $|\tilde{s}_q| < 1$ in order to ensure contractivity of the resulting operator. The codebook block D_k with minimal approximation error using quantized luminance coefficients, i.e., the codebook block D_k that minimizes the collage error $E(D, R)$ for range R given by

$$E(D, R) = \|R - (\tilde{s}_q D + \tilde{o}_q \mathbf{1})\|_2^2 \quad (3)$$

yields the fractal code for range R consisting of the corresponding domain address and isometry option as well as the luminance parameters \tilde{s}_q, \tilde{o}_q . By searching for the best fitting

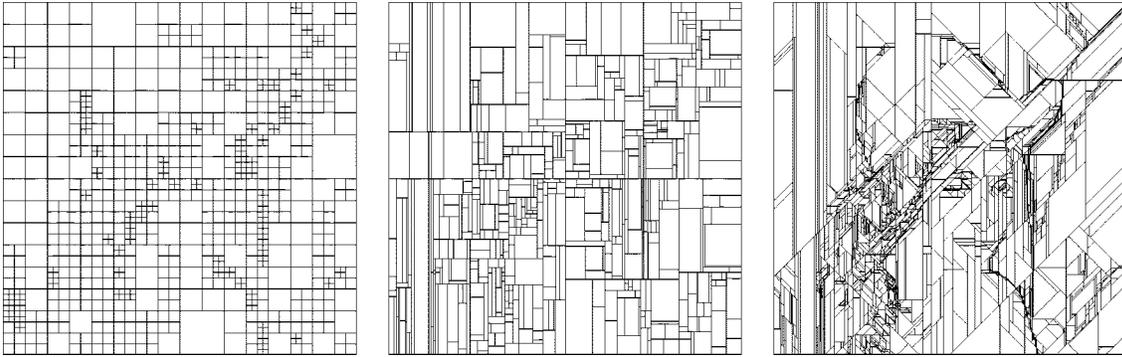


Fig. 1. Hierarchical approaches: quadtree, horizontal-vertical, polygonal.

codebook block for each range, one determines for each range the optimal geometric mapping (the domain address and the isometry option) and the optimal value transformation, and thus, together with the given partition one obtains the fractal code that minimizes the collage error. Thus, the collage coding method provides a fractal code for a given partition. In order to vary the rate, ranges can be split or merged, and for the newly constructed ranges a geometric mapping and a value transformation has to be determined. Various partitioning methods are discussed in the following section.

III. PREVIOUS WORK

In this section, we give a survey of the partitioning methods for fractal compression. We do not give a survey on other aspects of fractal coding and refer the reader to [6], [25], and [31].

The most basic class of partitions for fractal image compression are uniform partitions, by which we denote a partition consisting of *square* atomic blocks of size $j \times j$ pixels. Uniform partitions of other types can be defined by a regular tiling of the plane, but are rarely used for fractal image compression [26]. For a uniform partition based on square atomic blocks the block size j has to be specified; apart from that uniform partitions are image-independent. In the following we will focus on image-adaptive partitioning schemes.

A. Top-Down and Bottom-Up Hierarchical Approaches

Top-down quadtree-based fractal coders have been first presented in [27], [28]. Here, one starts with the whole image as a single range, and ranges are split into their four quadrants when the collage error for the range exceeds a given threshold. In [29] the splitting criterion based on the collage error is replaced by one based on the intensity variance of the range multiplied with the number of pixels in the range. Tests have shown a similar rate-distortion performance as with the collage error-based criterion.

Rate-distortion optimal quadtree partitions can be derived in a bottom-up fashion (cf. [30]); rate-distortion optimal quadtrees have been employed for fractal compression in [31, Sect. 4.3]–[33]. The cost for describing a quadtree partition is rather small, but there is only a limited adaptivity to the image content. A higher adaptivity can be obtained by additionally considering unions of quadrants after splitting a range as has

been done in the original work of Jacquin as well as, e.g., in [34], [10].

Fractal coding with horizontal-vertical (HV) partitions has been presented in [7]. The image is segmented into rectangles. If no domain match with acceptable collage error is found for a given rectangular range block, the block is split into two rectangles either by a horizontal or a vertical cut. The splitting is based on the most significant horizontal or vertical edge and also incorporates a rectangle degeneration prevention mechanism. For each split the direction and the splitting position have to be coded. The HV partition generally is more expensive in rate than the quadtree partition. In spite of the higher cost for the partition information the simulation results, as given in [7], show a considerable improvement over the quadtree-based fractal coder. Rate-distortion optimal HV-partitions were introduced in [35].

A step further in adaptivity is polygonal partitioning that is based on an image segmentation technique presented in [36] and applied for fractal encoding by Reusens [37]. It is similar to HV partitioning, but it also includes 45° and 135° cutting directions. Reusens uses the collage error as splitting criterion, and the direction and position of the cut is determined by a variance-based block-uniformity criterion.

Reusens has studied the polygonal partitions in order to analyze whether a higher degree in adaptivity improves the rate vs distortion behavior of fractal coding schemes. To this end Reusens has compared the polygonal partition-based fractal coder with a quadtree coder and an HV-coder which has been obtained by restricting the polygonal partition-based coder to horizontal and vertical cuts. From Reusens study one might conclude that better adaptivity does not improve rate-distortion performance. However, this is not the case as has been shown by Fisher's HV-coder as well as by our approach in this paper.

Example partitions for the hierarchical approaches are given in Fig. 1.

B. Split-and-Merge Approaches

Davoine *et al.* [38]–[40] advocate the use of Delaunay triangulation as partitioning method. The advantage of triangulations is the unconstrained orientation of edges. The Delaunay triangulation maximizes the minimal interior angle of its triangles which imposes some regularity. The method works as follows: one starts with a Delaunay triangulation of a set of regularly distributed points. Then, the partition is refined by splitting nonuniform triangles where uniformity is measured by the intensity

variance of the range. This splitting step is performed by adding an additional point at the barycenter of the nonuniform triangle and by recomputing the Delaunay triangulation for the new set of points. The splitting is stopped via a uniformity criterion. In the subsequent merging pass, a vertex p is removed, if all triangles with vertex p have approximately the same intensity mean value, and again the Delaunay triangulation of this new set of points is computed. In [39], [40] the merging of two triangles is also allowed when the resulting quadrilateral is convex and both triangles have more or less the same grey value distribution. The partition is coded by specifying the split and merge process. In order to deal with the large variety of different shapes, a wider class of geometric transformation is used instead of the usual affine one. The coding results presented in [40] are inferior to the results obtained with the HV approach.

The *Region-Based Fractal Coder Using Heuristic Search* of Thomas and Deravi [8] can also be regarded as a split-and-merge approach. One starts with a uniform partition, i.e., the image is split in atomic square blocks, e.g., of size 4×4 or 8×8 pixels. Then neighboring blocks are merged successively to build larger ranges of irregular shapes. Since one ends up with a few large ranges, there are only a few parameters to store. Thomas and Deravi give three methods differing in the level of sophistication. With the basic algorithm, for a seed atomic block an optimal domain match is sought. Then it is checked whether this transformation can be extended to a neighboring block. This extension step is stopped by a distortion criterion. Another seed is selected and the procedure goes on until the whole image is coded. The other two approaches also include some updating procedures and built-in competition.

C. Relation to Segmentation-Based Image Coding

When one requires that all scaling parameters of a block-based RB-operator have to be zero, fractal coding based on collage error minimization amounts to finding the optimal piecewise constant approximation for an image. Therefore, image coding by piecewise constant approximations can be regarded as a special case of fractal image compression. This relates fractal coding with adaptive partitions to *segmentation-based* or *second-generation* image coding [41], [42]. Second-generation coding methods take into account the human visual system and its property to differentiate between *edges* (features at a coarse scale) and *texture* (features at a fine scale). In segmentation-based image coding the image partition represents the edges whereas the ranges of the partition represent smooth image regions that can be sufficiently well approximated by low-order polynomials. We emphasize that this is generally not the case in fractal coding with adaptive partitions: an important edge may cross a range since for such a range a good domain might exist.

IV. REGION-BASED FRACTAL CODING

The partitioning scheme represents the main tool for the rate control in a fractal coder. As the results of previous fractal coders have indicated, fractal coding can profit from highly

flexible and image-adaptive partitioning schemes. However, the approaches presented so far suffered from the fact that a higher flexibility has led to excessive encoding times since during the encoding a large number of alternatives has to be tested. In this section we put forward our region-based fractal coder (RBFC) that generates highly image-adaptive partitions while keeping the computational costs of encoding low. With our RBFC the image partitions for the ranges are derived in a bottom-up fashion by first uniformly partitioning the image, and then by successively merging neighboring ranges. In Section IV-A we will focus on the region-merging strategy as well as on computational issues that make the coder time-efficient. In Section IV-B we will discuss an effective coding scheme for lossless coding of the resulting irregular partitions. As will be clear from the discussion of our merging strategy, the fractal codes obtained with the region-based fractal coder can be improved by performing an additional full domain search as a “postprocessing” step. A technique for performing such a full domain search efficiently is presented in Section IV-C.

The results presented in the following Section V will show that the region-based fractal coder outperforms the other pure fractal coders with respect to rate-distortion performance as well as with respect to encoding speed, and it will be shown that the additional full search leads only to a small improvement showing that our heuristic merging strategy performs very well.

A. Merging Criterion

Let us first elaborate on the difficulties involved in a region-merging approach to fractal coding. Assume that one is given a fractal code for an image, and two neighboring ranges have been selected for a merger. Then a fractal code for the new partition is obtained by keeping all parameters of the previous fractal code for the ranges not involved in the merger, and by replacing the parameters of the two selected ranges by a domain address, isometry option and luminance parameters for the new range. In order to determine a collage-optimal domain, a domain search for the new range is required. Now, when one likes to find in a given fractal code the neighboring range pair whose merger gives the least increase in collage error, one has to know for *each* neighboring range pair the best domain for the corresponding union of ranges. These considerations show that there are two related questions with regard to region-based fractal coding: how can one maintain a fractal code during the merging process, and how does one decide what neighboring range pair should be merged next? And, obviously, the main challenge of the region-merging approach to fractal coding lies in the problem of computational efficiency.

The region-merging process can be formalized as follows. Let f be an image of resolution n , and let j denote the atomic block size of the blocks of the initial uniform partition. The number of atomic blocks $n_{\mathcal{R}}$ is $(n/j)^2$. The region merging process corresponds to a sequence of $n_{\mathcal{R}}$ partitions $\mathcal{P}_0, \dots, \mathcal{P}_{n_{\mathcal{R}}-1}$, where the index gives the number of merging steps already performed, and \mathcal{P}_0 denotes the uniform partition with block size j . \mathcal{P}_i is constructed from \mathcal{P}_{i-1} , $0 < i < n_{\mathcal{R}}$, by merging two neighboring ranges of \mathcal{P}_{i-1} , thus partition \mathcal{P}_i has $n_{\mathcal{R}} - i$ ranges. For a region $P \in \mathcal{P}$ let us denote the vector corresponding to range

$f \upharpoonright P$ by R_P and the codebook of R_P by \mathcal{D}_{R_P} . One then can define the optimal collage error for image f and partition \mathcal{P} via

$$\mathcal{E}(\mathcal{P}) = \sum_{P \in \mathcal{P}} \left[\min_{D \in \mathcal{D}_{R_P}} \|R_P - (\tilde{s}_q D + \tilde{o}_q \mathbf{1})\|_2^2 \right] \quad (4)$$

where \tilde{s}_q, \tilde{o}_q are the optimal quantized luminance parameters for range R_P and codebook block D . At each step in the merging process we are interested in finding partition \mathcal{P}_i given the previous partition \mathcal{P}_{i-1} such that the increase in collage error

$$\mathcal{E}(\mathcal{P}_i) - \mathcal{E}(\mathcal{P}_{i-1}), \quad 0 < i < n_{\mathcal{R}}$$

is minimal, and the codebooks considered for determining the collage error (4) correspond to the full domain pools. Unfortunately, the computational costs for finding such a sequence of partitions depend on the sizes of the codebooks \mathcal{D}_{R_P} , and only very restricted sizes are manageable.

Therefore, the key idea of our approach is to use small codebooks that are constructed during the merging process using some knowledge that has already been accumulated in the process so far with respect to the self-similarities of the image. To this end we maintain at each step an *extended* fractal code that consists of

- an image partition \mathcal{P} ;
- for each range in the partition \mathcal{P} :
 - a list of $n_{\mathcal{B}}$ domain addresses $k_0, \dots, k_{n_{\mathcal{B}}-1}$;
 - optimal quantized luminance parameters \tilde{s}_q, \tilde{o}_q for domain k_0 .

The extended fractal code differs from the standard one because there are $n_{\mathcal{B}}$ domain addresses specified for each range instead of a single one. These addresses will indicate the domain pool considered for the corresponding range. $n_{\mathcal{B}}$ is a parameter of the method. For example, in the experiments reported in Section V $n_{\mathcal{B}}$ is set to 10. For the initial uniform partition the domain addresses $k_0, \dots, k_{n_{\mathcal{B}}-1}$ are set to the addresses of the $n_{\mathcal{B}}$ optimal or nearly optimal—in terms of the collage error $E(D, R)$ —domains for a range.⁴

After the initialization phase we begin to merge neighboring range pairs. In order to obtain a matching domain block for the union of two ranges, we consider only domains at positions that are given by the lists of domain addresses inherited from the two ranges. Of course, these domains have to be extended to match the size and shape of the new range. Specifically, the $n_{\mathcal{B}}$ addresses associated with a range $P \in \mathcal{P}$ represent the geometric mappings $g_P^{(k_0)}, \dots, g_P^{(k_{n_{\mathcal{B}}-1})}$. For a neighboring range pair P_1, P_2 we check the domains given by the geometric mappings $g_{P_1}^{(k_0)}, \dots, g_{P_1}^{(k_{n_{\mathcal{B}}-1})}, g_{P_2}^{(k_0)}, \dots, g_{P_2}^{(k_{n_{\mathcal{B}}-1})}$ applied to $P_1 \cup P_2$ (an example is given in Fig. 2). The rationale behind this approach is that a *good* domain for the new range should sufficiently well approximate both parts of the range, and, therefore, it will probably be an extension of one of the $2n_{\mathcal{B}}$ choices.

The new extended fractal code for partition \mathcal{P}_i is derived from the previous fractal code for partition \mathcal{P}_{i-1} by replacing the information for the two merged ranges with the information for the new range. The $n_{\mathcal{B}}$ domain addresses corresponding to the

⁴ k_0 denotes the address of the best matching domain.

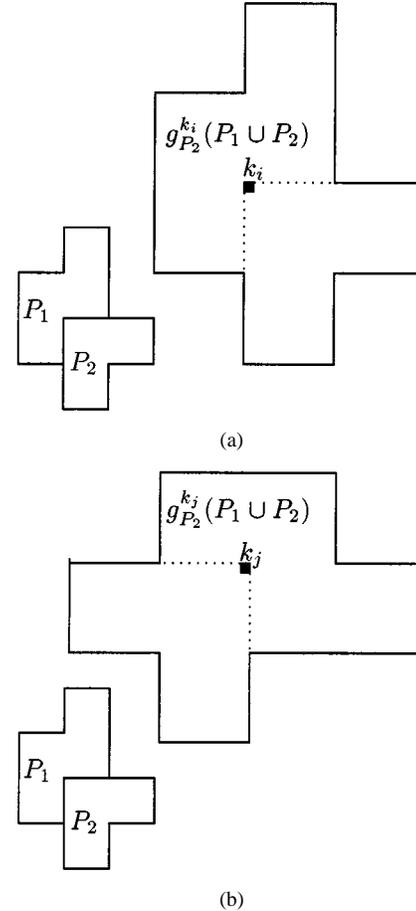


Fig. 2. (a) Example for the selection of a domain for the union of two neighboring ranges P_1, P_2 . When k_i is one of the $n_{\mathcal{B}}$ domain addresses for P_2 , the domain $g_{P_2}^{(k_i)}(P_1 \cup P_2)$ is considered for $P_1 \cup P_2$ and (b) considers the case of a rotation of 270° .

new range consist of the better half (with respect to the collage error) of the $2n_{\mathcal{B}}$ domain addresses considered for that range.

Overall, the process starts with a fractal encoding having a large rate and a small collage error. Each new partition has one less range. Thus, the rate decreases while the collage error increases. The process can be stopped when a given tolerance threshold for the collage error is exceeded or when the desired rate is obtained. Several partitions obtained with the region-merging process are depicted in Fig. 3.

Even though the codebooks used during the merging are small we propose two sub-algorithms to further reduce the encoding times; the first one is concerned with the initialization, the second one with the data handling during the region merging process.

In the initialization phase a fractal encoding for a uniform partition is sought. Here, an acceleration method can be employed, especially since all ranges have the same size. The method of Saupe [43] is particularly well suited for this case. We will briefly outline this method. For a range $R \in \mathbb{R}^m \setminus \text{span}\{\mathbf{1}\}$ and a codebook block $D \in \mathbb{R}^m \setminus \text{span}\{\mathbf{1}\}$ the collage error $\|R - (\tilde{s}D + \tilde{o}\mathbf{1})\|_2^2$ using *unquantized* luminance parameters can be expressed as

$$\left(\langle R, R \rangle - \frac{1}{m} \langle R, \mathbf{1} \rangle^2 \right) \cdot (1 - \langle \chi(D), \chi(R) \rangle^2).$$

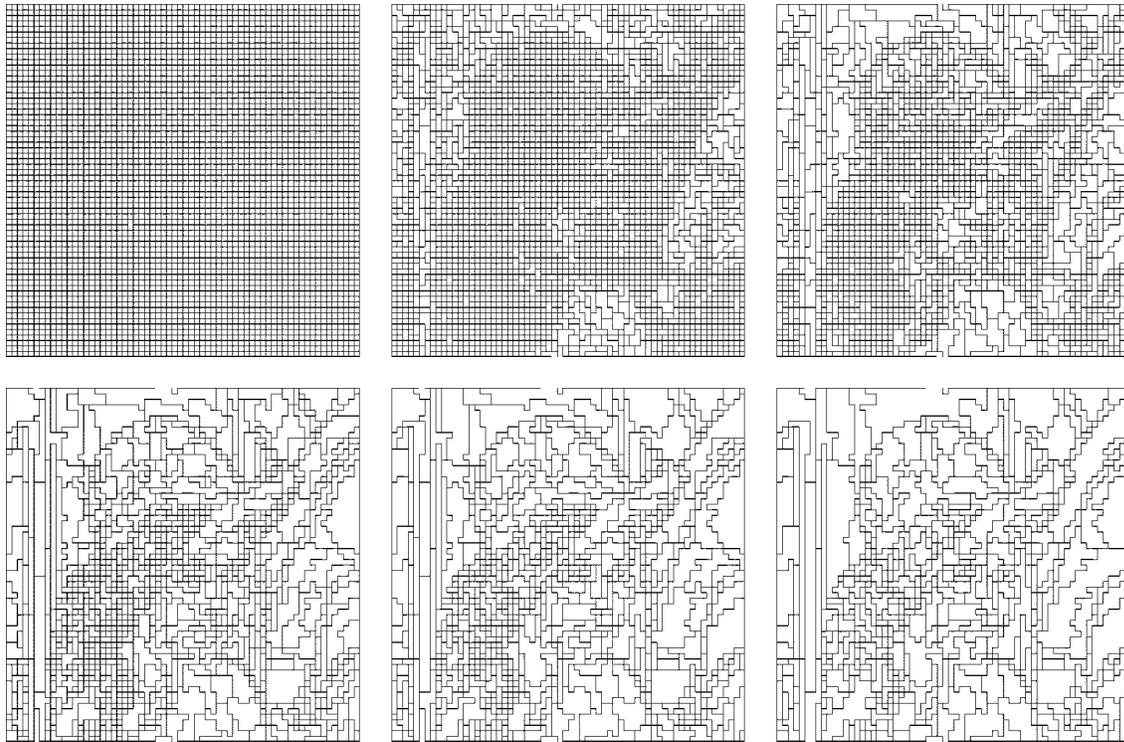


Fig. 3. Snapshots from a sequence of partitions derived with our region-based fractal coding method.

Here, $\chi(R)$, $\chi(D)$ denote the mean removed range and codebook block, respectively, normalized to unit variance. Thus, the best codebook block for range R is the one that maximizes $\langle \chi(D), \chi(R) \rangle^2$. Since

$$\|\chi(R) - \chi(D)\|_2^2 = 2 - 2\langle \chi(R), \chi(D) \rangle$$

the best codebook block D is the one that minimizes the Euclidean distance $\|\chi(R) - \chi(D)\|_2$ or $\|\chi(R) - (-\chi(D))\|_2$, respectively, (note that $\chi(-D) = -\chi(D)$). Therefore, the codebook search is reduced to a nearest neighbor search in the Euclidean space. Since all ranges are compared against the same codebook blocks, a fast nearest neighbor search based on binary space partitions (*kd-trees*, see [44]) leads to significant acceleration factors. To further accelerate the initialization phase, one only performs an *approximate* nearest neighbor search [45], i.e., one only queries for points whose distance is at most $(1 + \epsilon)$ times larger than the distance of the closest neighbor, where ϵ is the parameter of the method. Even for large values of ϵ , e.g., $\epsilon = 2.5$, the method produces excellent results. Because of the approximate nearest neighbor search, and because of the fact that the optimization is with respect to unquantized luminance parameters, we are not guaranteed to obtain the optimal initial fractal code. However, experimental results have shown that this method leads to a negligible loss in quality while providing a significant speedup [46].

In the merging phase, one has to consider all neighboring range pairs, and for a merger a range pair is chosen that gives the minimal increase in collage error. To support this task, a *priority queue with lazy evaluation* is maintained in which all neighboring range pairs are listed. The entries of the priority queue are sorted by the increase of collage error resulting from

a merger of the range pair. One, therefore, can easily extract the range pair that has to be merged next. After picking the top element, the priority queue has to be updated. The following key observation makes the updating simple and efficient. We can delay the recomputation of the collage error of range pairs for which one member has been merged in the previous step since the error for such a pair can only become larger. Because of this monotonicity property, it is sufficient to perform the recomputation when such an outdated pair is extracted as top element of the priority queue. In this case the correct collage error has to be computed and the element has to be reinserted into the priority queue. Thus, when extracting the top element of the queue one only has to check whether the involved ranges are still *valid*, i.e., whether they still exist in the current partition. An outline of the entire procedure—where the stopping criterion is given by the number of ranges—is presented as Algorithm 1.

Algorithm 1 Region-based fractal encoding
Input: image $f \in \mathbb{R}^{n \times n}$, number of domains per range n_B , number of ranges in the final partition k , atomic block size j .

- Find extended fractal code for the uniform partition with atomic block size j .
- Initialize priority queue (PQ): put each neighboring range pair in the PQ, sorted by corresponding increase of collage error.

```

while current number of ranges >  $k$  do
  • Extract top element of PQ.
  if involved ranges are still valid
  then

```

- Update partition: merge range pair.
- Update fractal code: determine n_B domains for new range.
- Current number of ranges \leftarrow current number of ranges -1 .

else

- Update range pair: compute approximation error and reinsert in PQ.

end if

end while

Output: partition, opt. luminance parameters, domain addresses, and isometry options.

				3	1	3	1
				2	0	1	3
				3	1	1	3

Fig. 4. Example of a region edge map.

NW	N	NE
W	ω	

Fig. 5. Context template for the encoding of symbol ω .

B. Coding of Irregular Partitions

We now discuss efficient encoding of partitions generated by the region-based fractal coder. We have focused on two different methods to encode the configuration of range boundaries: *chain codes* and *region edge maps*. With chain codes the boundaries between the ranges are traced resulting in a stream of symbols that represent a starting point and actions like *turn right*, *go ahead*, and so on. The resulting stream of symbols is entropy coded. The chain codes have been used to digitally represent curves since the publication of the paper of Freeman [47]. When tracing the boundary no edge should be traced twice, and, since in general there is no single path through the boundary graph that visits each edge exactly once, new starting points should be given in an efficient (implicit) fashion. Details for constructing efficient chain codes are given in, e.g., [11]. Since the coding results obtained with region edge maps outperform the chain coding method, we will focus on region edge map encoding.

With a region edge map two bits are specified for each atomic block of the uniform partition from which the irregular partition has been derived. The two bits indicate whether there is an edge, i.e., a range boundary, to the north of the block and whether there is an edge to the west of the block. For instance, we let 0 stand for no edge, 1 for a north edge, 2 for a west edge, and 3 for a north and a west edge. An example is given in Fig. 4. An efficient method for encoding these region edge maps via context modeling and adaptive arithmetic coding has been proposed by Tate [48] and applied to the coding of palette images [49].

The context used for the encoding of region edge maps depends on the neighbors to the west, north, northwest, and northeast of the symbol to be coded (using the normal raster scan order all those neighbors are coded *before* the current symbol), see Fig. 5. Each of the 256 different symbol combinations in $\{0, 1, 2, 3\}^4$ is considered to be a context for symbol ω . One can easily see that from this choice of contexts one can infer some information for the symbol to be coded: for example, when the symbols to the north and to the west of the current symbol are both zero indicating that there are no north and west edges for these atomic blocks, one immediately knows that the current symbol has to be 0 or 3 since the range boundaries constitute a connected graph.⁵ The number of contexts is small and, thus,

⁵The connectedness can be violated at the image boundaries because of wrap-around ranges.

there is no context dilution problem for encoding partitions of images of typical size, say 512×512 .

The adaptive arithmetic coder is initialized by setting the frequency count for all events that can actually occur to 1. Only events that cannot occur get a frequency count of 0. The frequency count for an event is increased by 4 each time the event is observed. This incrementation step size has been experimentally determined to be the best choice for our application.

Within MPEG-4 several shape coding techniques have been studied where shapes are given by means of contours, bitmaps, or chroma-keying [50]. These methods are not directly applicable to encoding the partitions in region-based fractal image compression where a large set of regions must be encoded as a whole in contrast to individual video objects in MPEG-4. However, our chain codes in [11] and the region edge maps discussed above are closely related to the contour and bitmap coding approaches studied in the MPEG group. As mentioned above, the region edge maps were more efficient than the chain codes even when the latter were equipped with sophisticated context modeling. This is in line with the results of the MPEG-4 shape coding evaluation where the context-based arithmetic bit-map coder (CAE) won over the contour coders. Intuitively, the reason may lie in the fact that regions are twodimensional. When we encode boundaries which are onedimensional, we may have less coherence in the symbol strings than there is coherence within a bitmap or a region edge map. In other words, the context-based coding can be expected to be more powerful when applied to region edge maps rather than contour chain codes.

C. Additional Full Search

For a fractal code derived with our region-based fractal coder each domain-range assignment is optimal only with respect to the restricted domain pool scheme presented in Section IV-A, i.e., each domain represents the best out of a set of heuristically chosen n_B domains. One can expect better matching domains by considering larger domain pools. In the following we will focus on the problem of how to determine for a partition derived by our RBFC a fractal code that assigns to each range an optimal domain of the corresponding full domain pool. Thus, we need a fractal coder that is time-efficient in the presence of large irregularly shaped ranges. With such a coder one can improve a fractal code obtained by our region-based fractal coder, and, in turn, can evaluate the quality of our heuristic merging strategy.

We propose a coder based on the fast cross-correlation property of the Fourier transform. This coder exploits the codebook coherence of *full* codebooks, i.e., the codebooks that correspond to full domain pools [51].

In order to find the best matching domain for a range, one has to determine for each domain the optimal luminance parameters. To this end one has to compute Euclidean inner products $\langle D, D \rangle$, $\langle D, \mathbf{1} \rangle$, $\langle D, R \rangle$, $\langle R, \mathbf{1} \rangle$, and $\langle R, R \rangle$, where $R \in \mathbb{R}^m$ denotes a range vector and $D \in \mathbb{R}^m$ a codebook vector, since

$$\begin{aligned} \|R - (sD + o\mathbf{1})\|_2^2 \\ = \langle D, D \rangle s^2 + 2\langle D, \mathbf{1} \rangle o + mo^2 - 2\langle R, D \rangle s \\ - 2\langle R, \mathbf{1} \rangle o + \langle R, R \rangle. \end{aligned}$$

For an image of resolution n , n even, the full domain pool for a range consists of $8 \cdot n^2/4$ domains. For simplicity, we will first assume that only one isometry option—no rotation, no flip—is allowed, i.e., the domain pool consists of $n^2/4$ domains. The corresponding full codebook consists of all blocks of the downsampled image (obtained by pixel averaging over disjoint 2×2 blocks) with the same size and shape as the range; thus, the codebook blocks can be indexed via D_{n_1, n_2} , $0 \leq n_1, n_2 < n/2$. For large ranges the computational costs of the domain search is dominated by the computation of the inner products $\langle D_{n_1, n_2}, R \rangle$, $0 \leq n_1, n_2 < n/2$. These inner products $\langle D_{n_1, n_2}, R \rangle$, $0 \leq n_1, n_2 < n/2$ are the cross correlation of the corresponding range with the downsampled image. Let an image f of resolution n , n even, be given as well as an appropriate partition \mathcal{P} , and let $h_1 \in \mathbb{R}^{n/2} \times \mathbb{R}^{n/2}$ represent the downsampled image via

$$h_1(k_1, k_2) = \sum_{x=2k_1}^{2k_1+1} \sum_{y=2k_2}^{2k_2+1} f\left(\frac{x}{n}, \frac{y}{n}\right).$$

Let $h_2 \in \mathbb{R}^{n/2} \times \mathbb{R}^{n/2}$ denote a *zero-padded* range $f \upharpoonright P$, $P \in \mathcal{P}$, given by

$$h_2(k_1, k_2) = \begin{cases} f\left(\frac{k_1}{n}, \frac{k_2}{n}\right), & \left(\frac{k_1}{n}, \frac{k_2}{n}\right) \in P \\ 0, & \text{otherwise} \end{cases}.$$

Now, by calculating the cross correlation $h_2 \circ h_1$, i.e.,

$$\begin{aligned} h_2 \circ h_1(n_1, n_2) &= \sum_{k_1, k_2=0}^{n-1} h_2(k_1, k_2) \\ &\quad \cdot h_1((k_1 + n_1) \bmod n, (k_2 + n_2) \bmod n) \end{aligned}$$

one obtains all inner products between a range and the codebook blocks of the corresponding full domain pool. It is well-known that the cross correlation can be efficiently computed in the Fourier domain for ranges that are not too small. This leads to the following computation scheme for the inner products between a range and all codebook blocks of the corresponding domain pool:

- 1) compute the 2D-FFT of the downsampled image h_1 ;
- 2) build zero-padded range h_2 ;
- 3) compute the 2D-FFT of h_2 ;
- 4) perform the complex conjugate of previous transform;
- 5) do the (pointwise) complex multiplication of both transforms;
- 6) do the inverse FFT of the result.

Moreover, the calculation of $\langle D_{n_1, n_2}, D_{n_1, n_2} \rangle$, $\langle D_{n_1, n_2}, \mathbf{1} \rangle$ for $0 \leq n_1, n_2 < n/2$ can be accelerated by the same cross correlation technique. The products of type $\langle D, \mathbf{1} \rangle$ are obtained by the cross correlation of the downsampled image with a “range” where all intensities are set to unity (called the *range shape matrix*). The sum of squares $\langle D, D \rangle$ is computed in a similar way by squaring all intensities in the downsampled image before cross correlation.

In order to incorporate all isometry options, one can take advantage of direct methods to obtain the Fourier transforms for rotated and reflected images. Therefore, there is no need to compute eight forward transforms for the eight isometric versions of the downsampled image. Instead, one forward transform is computed and the Fourier transforms of the rotated and reflected downsampled image are obtained directly in the Fourier domain.

The proposed FFT-based coder is particularly suited for irregular partitions since it requires only *zero-padding* of a range. After this, the procedure does not depend on the range shape. This approach shows significant speedups for ranges with more than 256 pixels compared to an optimized direct computation. For images of size 512×512 and a range with 256 pixels the speedup factor is about 6.6; for a range consisting of about 1000 pixels the speedup factor is 25.

V. RESULTS

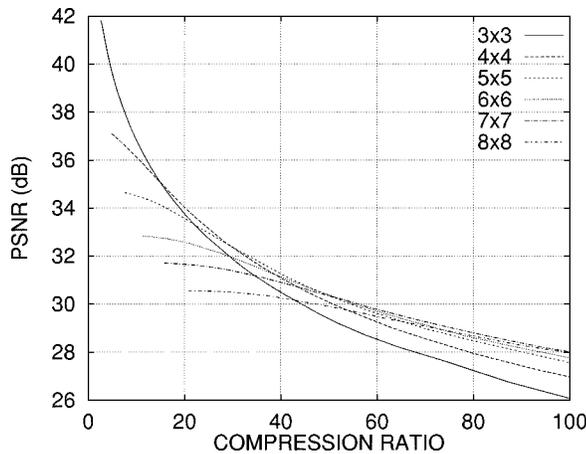
In this section, we present coding results obtained with our region-based fractal coder that utilizes the region edge map encoding scheme for the coding of the resulting partitions. The results are given for the standard test images Lenna and boat.

We have chosen the following parameter settings for our tests. In the initialization phase, the image is uniformly partitioned into blocks of size $j \times j$; we have performed experiments for $j = 3, 4, 5, 6, 7, 8$. The domain pool considered for the initial fractal code consists of the blocks of the uniformly partitioned image with atomic block size $2j \times 2j$ plus the isometric versions thereof. In our experiments the number n_B of domain addresses per range has been set to 10. For the approximate nearest neighbor search used as acceleration method for the initialization phase the allowable approximation error ϵ (cf. Section IV-A) has been set to $\epsilon = 2.5$. The quantization for the luminance parameters is performed as it is done in Fisher’s quadtree-based fractal coder [6] with 2^5 levels for the scaling factor and 2^7 levels for the offset. We allow wrap-around ranges as well as wrap-around domains to occur during the merging process.

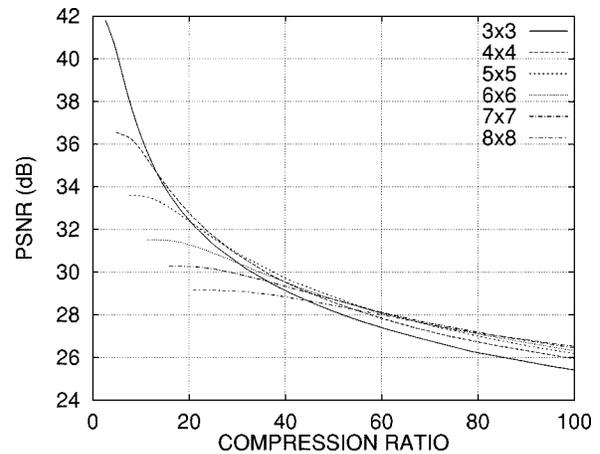
In Fig. 6(a), coding results for compression ratios up to 100 : 1 are presented for the 8-bit Lenna image of size 512×512 pixels.⁶ Compression ratio vs PSNR curves are given which correspond to region merging processes that are initialized with uniform partitions varying with respect to the atomic block size.⁷ The atomic block size can be chosen in order to control the regularity of the resulting partitions. A small atomic block size is advantageous for very-high fidelity encodings while larger atomic block sizes should be selected for high compression ratios.

⁶The image was obtained from the Waterloo BragZone [52].

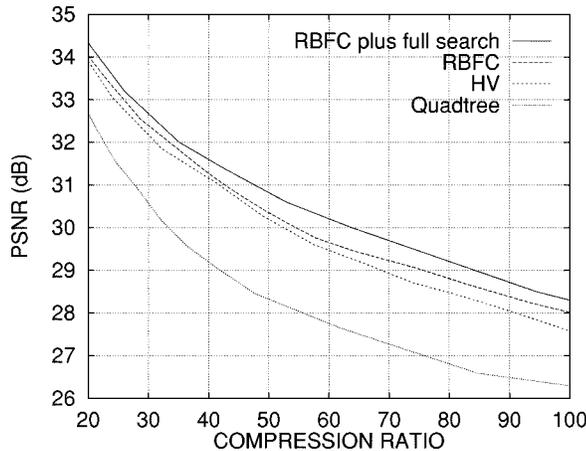
⁷At the right and bottom image boundaries the block sizes are adapted in order to match the given image size.



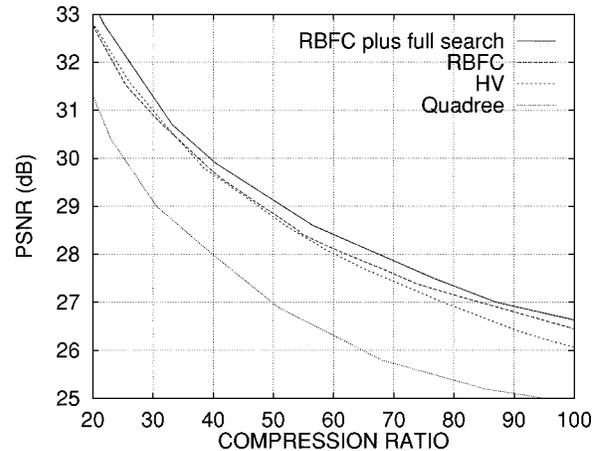
(a)



(a)



(b)



(b)

Fig. 6. (a) Compression ratio versus PSNR curves for the Lenna 512×512 image obtained with the RBFC scheme starting with various atomic block sizes and (b) comparison of the RBFC approach with additional full search, the standard RBFC approach, the HV scheme, and the quadtree-based fractal coder.

Fig. 7. (a) Compression ratio versus PSNR curves for the boat 512×512 image obtained with the RBFC scheme starting with various atomic block sizes and (b) comparison of the RBFC approach with additional full search, the standard RBFC approach, the HV scheme, and the quadtree-based fractal coder.

We compare the results of the RBFC approach (with and without additional full search) with the results obtained by Fisher's HV⁸ and quadtree-based coders; the HV coder is considered as a "reference coder" for the current state-of-the-art in fractal image compression. Results for the 512×512 Lenna image are given in Fig. 6(b). Here, we focus on high compression ratios, i.e., ratios between 20:1 and 100:1 since for those ratios fractal coding methods have shown their competitive edge. All options for the HV coder have been enabled⁹ and the decoding has been done without postprocessing. The options for the quadtree coder has been set to `-m 4 -M 7 -w 512 -d 1 -D 2 -f -F`¹⁰; again no postprocessing has been performed after the decoding. The RBFC yields a slightly better rate-distortion curve compared to the HV method when for each compression ratio the PSNR value obtained with the best

⁸We thank Y. Fisher for supplying us with a Sun-executable of his HV-coder.

⁹Fisher's optimization level 0 (cf. [6, p. 127]): "Compare each range with each possible domain in the image, using every possible symmetry operation. The ratio of the lengths of the sides of the domain and range blocks are 2×2 , 2×3 , 3×2 , and 3×3 ."

¹⁰The options are outlined in [6, p. 260, 261]; the ranges are of sizes 4×4 , 8×8 , 16×16 , and 32×32 , and a full search is performed.

atomic block choice is taken into account. The RBFC approach with the additional full search can achieve coding gains of 0.5–0.8 dB for compression ratios between 20:1 and 100:1 over the HV-results. It is obvious from Fig. 6(b) that a higher flexibility of the partitioning method gives significant coding gains over the more rigid quadtree-based approach. Fig. 7(a) and (b) present corresponding results for the 512×512 boat image.

Typical computing times (CPU seconds) for a single ratio-PSNR curve are about 170 s on a Sparc processor at 170 MHz. Let us compare the encoding times of the RBFC approach and the HV coder [7] for a Lenna encoding at a compression ratio of 65:1. The RBFC approach requires about 130 s and results in a 29.5 dB PSNR. The HV coder with a standard full search (level 3) needs about 250 s and results in a 28.8 dB PSNR. Enabling all options for the HV coder (level 0) leads to encoding times of about 3700 s and 29.2 dB PSNR. The RBFC approach with additional full search has been implemented with hardware-optimized Fourier transforms on an SGI using a MIPS R5000 processor at 150 MHz. A 65:1 Lenna encoding costs about 2800 s and results in a 29.9 dB

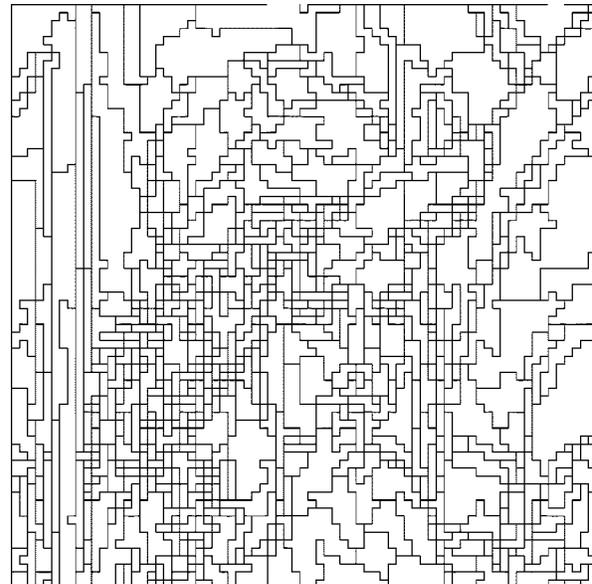


(a)

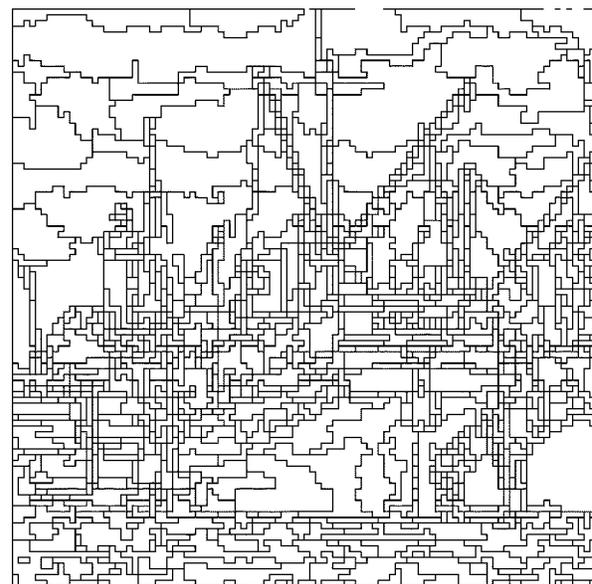


(b)

Fig. 8. (a) Decoded Lenna 512×512 image at compression ratio 64 : 1 and 30 dB PSNR and (b) decoded boat 512×512 image at compression ratio 40 : 1 and 29.8 dB PSNR.



(a)



(b)

Fig. 9. (a) Partition corresponding to Fig. 8(a) (800 ranges) and (b) partition corresponding to Fig. 8(b) (1200 ranges).

PSNR. To summarize, the RBFC method slightly outperforms the quality of the best HV encodings and is significantly faster than the HV coder. The additional full search obtains an additional gain of about 0.4 dB PSNR, however, this increases encoding times by about 2670 CPU seconds.

Figs. 8(a) and 9(a) show the decoded Lenna image at a compression ratio of 64 : 1 and the corresponding partition. The irregular structure of the partition results in less visible blocking artifacts in the decoded image compared to quadtree-based fractally coded images. Figs. 8(b) and 9(b) show the decoded boat image at a compression ratio of 40 : 1 and the corresponding partition. The reconstruction quality and the crispness of the images are visually pleasing considering the high compression ratios. However, while the reconstruction of

the edges is in general excellent, high-frequency texture details, e.g., the structure of the sand in the boat image, cannot be reconstructed very well.

In a fractal code obtained with the region-based fractal coder, a much larger part is used to specify the partition as in the case, e.g., of a quadtree-based fractal code. Fig. 10(a) and (b) give the relative and absolute partition coding costs for the Lenna encodings based on an initial uniform partition with atomic block size 4×4 . For a compression ratio of 66 : 1 the RBFC scheme results in a fractal code of which about 44% specify the partition. By contrast, at the same compression ratio the fractal code derived with the quadtree-based fractal coder needs only 3.5% of the total code size to define the partition.

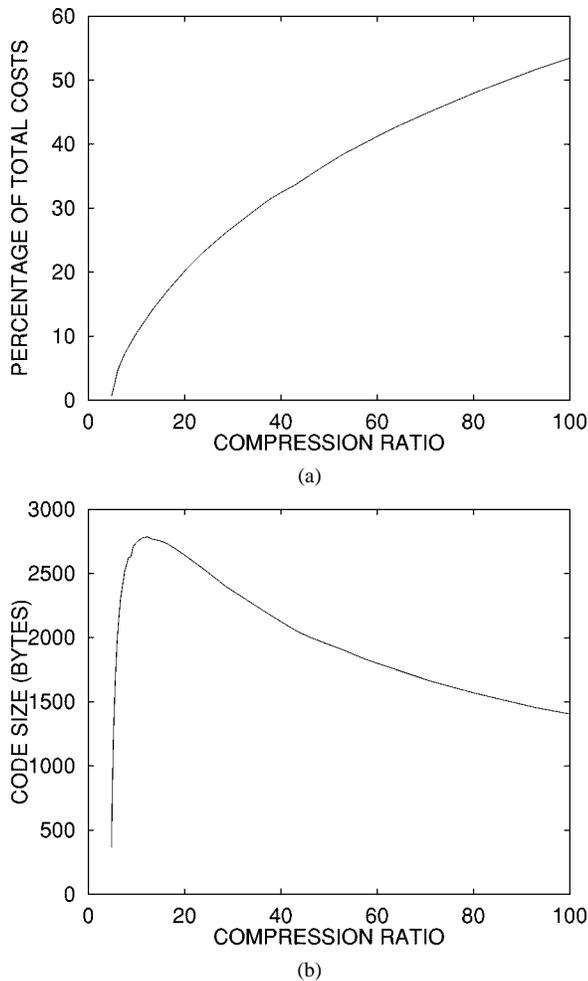


Fig. 10. (a) Part of fractal code that is used to specify the partition (in percent) and (b) absolute costs of the partition information; the data stems from the encodings of the 512×512 Lenna image starting with a uniform partition with atomic block size 4.

VI. ADDITIONAL INVESTIGATIONS

A. R/D-Based Region Merging

With the RBFC method the merging criterion is based on the increase in distortion that results from merging two ranges. Additionally, one may take into account the corresponding change in rate. In this case the Lagrangian cost function

$$\mathcal{E}(\mathcal{P}) + \lambda \cdot \text{rate}(\mathcal{P}) \quad (5)$$

is motivated where $\lambda \in \mathbb{R}^+$ is the scale parameter and $\text{rate}(\mathcal{P})$ denotes the sum of the coding costs for the region edge map encoded partition and the transformation parameters of the fractal encoding. Thus, at each step the neighboring range pair is merged that gives the largest decrease with respect to the cost function (5). The merging process continues until a further decrease of the cost function is no longer possible. Then the parameter λ is increased allowing merge operations again, and the procedure is iterated. However, the computational costs for the procedure are prohibitive as can be seen as follows. For each pair of neighboring ranges one has to keep track of the change in distortion and rate resulting from merging the two ranges. After merging a range pair all those rates change because the partition has changed. As a consequence the lazy evaluation in

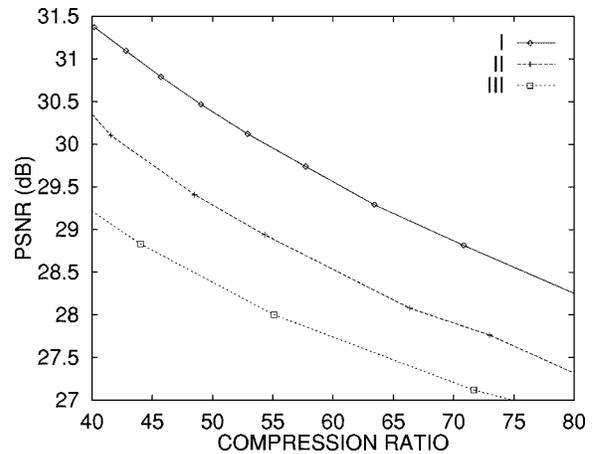


Fig. 11. Comparison of the rate-distortion performance of I) our RBFC approach, II) the region-based coder that uses R/D-based region merging, and III) Fisher's quadtree-based fractal coder.

the priority queue of the range block pairs is no longer justified and, thus, all the rate changes need to be computed by entropy encoding a whole partition for each range block pair in the priority queue. Therefore, this approach is not practical, not even for a single test in order to evaluate the losses caused by not using the strict rate-distortion criterion for the merging step. Thus, it is reasonable to relax the R/D criterion by replacing the actual coding costs with some rate estimator that can be more easily calculated. Our region-based fractal coder from Section IV treats the rate just as a constant. In this section, we consider a simple rate estimator as a first approximation. Motivated by the image segmentation approaches of Mumford and Shah [53] and Morel and Solimini [54] we have employed the Euclidean length $\ell(\mathcal{P})$ of the partition boundaries as an estimator for the actual partition coding costs. However, the results obtained with this approach are inferior to the results obtained with the pure distortion-based approach (cf. Fig. 11 and [13]). Thus, a more accurate yet simple estimator is needed. This warrants further investigation relevant also for other segmentation-based image coding schemes.

B. Probabilistic Merging

The RBFC approach has outperformed the method of Thomas and Deravi as well as the state-of-the-art top-down fractal coders. Nevertheless, since finding the optimal partition for an image at a given rate represents a difficult problem (as indicated in Section I), it is difficult to answer the question of how close the RBFC scheme gets to optimality. In order to obtain an idea of the potential of highly image-adaptive partitions for fractal coding, one can use a "screening" technique like our *evolutionary fractal coding* approach [11].

Here, a *generation* consists of N_p extended fractal codes (normally with differing partitions). A new generation is constructed from the previous one as follows. Each of the N_p extended fractal code is duplicated σ times. Then, in each of the σN_p extended fractal codes one neighboring range pair is randomly selected and merged. Finally, the N_p best (with respect to the collage error) extended fractal codes are kept to form the new generation. The domain search for a newly merged range is identical

to the one described for our region-based fractal coder. As initial generation, N_p copies of the initial code of our region-based fractal coder are used.

In the experiments we have set N_p to 10 and σ to 20; the number of domain addresses n_B maintained for each range in an extended fractal code has been set again to 10. For example, for the Lenna 512×512 image we have obtained slightly better rate-distortion curves of about 0.3 dB compared to the deterministic RBFC method. Of course, the probabilistic method spends much more time in the region merging phase, i.e., the region merging takes about 100 times longer compared to the deterministic case.

These results indicate that with optimal partitions one would get some improvement in the rate-distortion behavior, however, it appears, that the improvement would not be very large. This observation backs our heuristic region-merging strategy.

VII. SUMMARY AND CONCLUSIONS

In this paper, we have presented a fractal coder that is based on highly image-adaptive partitions. The partitions are derived bottom-up via region merging. At each step the range pair that gives the least increase in collage error is merged. This approach is made computationally feasible by only performing a domain search in small domain pools that are constructed during the merging process. These domain pools represent self-similarities in the image learned in the initialization phase and during the merging process. Time efficiency is obtained by 1) using an initialization technique that applies nearest-neighbor search techniques and 2) by lazy evaluation of the priority queue of range pairs during the merging phase.

Our region-based fractal coder clearly outperforms the current state-of-the-art purely fractal coders with respect to rate vs distortion performance as well as time efficiency.

REFERENCES

- [1] M. F. Barnsley and A. D. Sloan, "Chaotic compression," *Comput. Graph. World*, Nov. 1987.
- [2] —, "A better way to compress images," *Byte*, pp. 215–223, Jan. 1988.
- [3] A. E. Jacquin, "A fractal theory of iterated Markov operators with applications to digital image coding," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, Aug. 1989.
- [4] —, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Processing*, vol. 1, pp. 18–30, 1992.
- [5] T. Asano, D. Z. Chen, N. Katoh, and T. Tokuyama, "Polynomial-time solutions to image segmentation," in *Proc. 7th Annu. ACM-SIAM Symp. Discrete Algorithms*, 1996, pp. 104–113.
- [6] Y. Fisher, Ed., *Fractal Image Compression—Theory and Application*. New York: Springer, 1994.
- [7] Y. Fisher and S. Menlove, "Fractal encoding with HV partitions," in *Fractal Image Compression—Theory and Application*, Y. Fisher, Ed. New York: Springer, 1994.
- [8] L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE Trans. Image Processing*, vol. 4, pp. 832–838, June 1995.
- [9] G. M. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Trans. Image Processing*, vol. 7, pp. 141–154, Feb. 1998.
- [10] K. U. Barthel, "Festbildcodierung bei niedrigen Bitraten unter Verwendung fraktaler Methoden im Orts- und Frequenzbereich," Ph. D. dissertation, Institut für Fernmeldetechnik, TU Berlin, W&T Verlag, Berlin, 1996.

- [11] D. Saupe and M. Ruhl, "Evolutionary fractal image compression," in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, 1996, pp. 129–132.
- [12] M. Ruhl, H. Hartenstein, and D. Saupe, "Adaptive partitionings for fractal image compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Santa Barbara, CA, 1997, pp. 310–313.
- [13] H. Hartenstein and D. Saupe, "Cost-based region growing for fractal image compression," in *Proc. IX Eur. Signal Processing Conf. EU-SIPCO*, Sep. 1998, pp. 2313–2316.
- [14] A. H. Read, "The solution of a functional equation," in *Proc. R. Soc. Edinburgh*, vol. 63, sec. A, 1951–1952, pp. 336–345.
- [15] M. Bajraktarevic, "Sur une equation fonctionnelle," *Glasnik matemacko-fizicki i astronomski/Društvo Matematicara i Fizicra S.R. Hrvatske*, ser. II, vol. 12, pp. 201–205, 1957.
- [16] S. Dubuc, "Functional equations connected with peculiar curves," *Lecture Notes Math.*, vol. 1163, pp. 33–40, 1990.
- [17] P. R. Massopust, *Fractal Functions, Fractal Surfaces, and Wavelets*. New York: Academic, 1994.
- [18] T. Bedford, F. M. Dekking, and M. S. Keane, "Fractal image coding techniques and contraction operators," *Nieuw Arch. Wisk.*, vol. 10, pp. 185–217, 1992.
- [19] H. G. Heuser, *Functional Analysis*. Chichester, U.K.: Wiley, 1982.
- [20] D. M. Monro, "A hybrid fractal transform," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 5, 1993, pp. 169–172.
- [21] H. Hartenstein, "Topics in Fractal Image Compression and Near-Lossless Image Coding," Ph.D. dissertation, Institut für Informatik, Univ. Freiburg, 1998.
- [22] Z. Baharav, D. Malah, and E. Karmin, "Hierarchical interpretation of fractal image coding and its applications," in *Fractal Image Compression—Theory and Application*, Y. Fisher, Ed. New York: Springer, 1994.
- [23] M. Ruhl and H. Hartenstein, "Optimal fractal coding is NP-hard," in *Proc. IEEE Data Compression Conf.*, J. Storer and M. Cohn, Eds. Snowbird, UT, 1997, pp. 261–270.
- [24] M. F. Barnsley, *Fractals Everywhere*. New York: Academic, 1988.
- [25] D. Saupe, R. Hamzaoui, and H. Hartenstein, "Fractal image compression: An introductory overview," in *Fractal Models for Image Synthesis, Encoding, and Analysis*, D. Saupe and J. Hart, Eds. New Orleans, LA, 1996.
- [26] F. Mendivil, "The application of a fast nonseparable discrete periodic wavelet transform to fractal image compression," in *Proc. Fractals Eng.*, Delft, The Netherlands, 1999.
- [27] Y. Fisher, E. W. Jacobs, and R. D. Boss, "Fractal image compression using iterated transforms," in *Image and Text Compression*, J. Storer, Ed. Norwell, MA: Kluwer, 1992.
- [28] T. Bedford, F. M. Dekking, M. Brewer, M. S. Keane, and D. van Schooneveld, "Fractal coding of monochrome images," *Signal Process.*, vol. 6, pp. 405–419, 1994.
- [29] D. Saupe and S. Jacob, "Variance-based quadrees in fractal image compression," *Electron. Lett.*, vol. 33, no. 1, pp. 46–48, 1997.
- [30] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. Image Processing*, vol. 3, pp. 327–333, May 1994.
- [31] N. Lu, *Fractal Imaging*. New York: Academic, 1997.
- [32] R. Hamzaoui and D. Saupe, "Combining fractal image compression and vector quantization," *IEEE Trans. Image Processing*, vol. 9, pp. 197–208, Feb. 2000.
- [33] G. Melnikov and A. K. Katsaggelos, "A non uniform segmentation optimal hybrid fractal/DCT image compression algorithm," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, 1998.
- [34] T. Fuchigami, S. Yano, T. Komatsu, and T. Saito, "Fractal block coding with cost-based partitioning," in *Proc. Int. Picture Coding Symp.*, Melbourne, Australia, 1996, pp. 335–340.
- [35] D. Saupe, M. Ruhl, R. Hamzaoui, L. Grandi, and D. Marini, "Optimal hierarchical partitions for fractal image compression," in *Proc. ICIP-98 IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998.
- [36] X. Wu and C. Yao, "Image coding by adaptive tree-structured segmentation," in *Proc. IEEE Data Compression Conf.*, J. A. Storer and J. H. Reif, Eds. Snowbird, UT, 1991, pp. 73–82.
- [37] E. Reusens, "Partitioning complexity issue for iterated function systems based image coding," in *Proc. VIIth Eur. Signal Processing Conf. EU-SIPCO*, vol. 1, Edinburgh, U.K., 1994, pp. 171–174.
- [38] F. Davoine and J. M. Chassery, "Adaptive Delaunay triangulation for attractor image coding," in *Proc. 12th Int. Conf. Pattern Recognition*, Jerusalem, Israel, 1994.
- [39] F. Davoine, J. Svensson, and J. M. Chassery, "A mixed triangular and quadrilateral partition for fractal image coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Washington, DC, 1995, pp. 284–287.

- [40] F. Davoine, G. Robert, and J. M. Chassery, "How to improve pixel-based fractal image coding with adaptive partitions," in *Fractals in Engineering*, J. L. Véhel, E. Lutton, and C. Tricot, Eds. London, U.K.: Springer-Verlag, 1997, pp. 292–306.
- [41] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proc. IEEE*, vol. 73, pp. 549–574, Apr. 1985.
- [42] H. Radha, M. Vetterli, and R. Leonardi, "Image compression using binary space partitioning trees," *IEEE Trans. Image Processing*, vol. 5, pp. 1610–1624, Dec. 1996.
- [43] D. Saupe, "Accelerating fractal image compression by multi-dimensional nearest neighbor search," in *Proc. IEEE Data Compression Conference*, J. Storer and M. Cohn, Eds. Snowbird, UT, 1995, pp. 222–231.
- [44] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.
- [45] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu, "An optimal algorithm for approximate nearest neighbor searching," in *Proc. 5th Annu. ACM-SIAM Symp. Discrete Algorithms*, 1994, pp. 573–582.
- [46] D. Saupe, "Fractal image compression via nearest neighbor search," in *Conf. Proc. NATO ASI Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.
- [47] H. Freeman, "On the coding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 260–268, 1961.
- [48] S. R. Tate, "Lossless compression of region edge maps," Dept. Comput. Sci., Duke Univ., Durham, NC, Tech. Rep. CS-1992-9, 1992.
- [49] P. J. Ausbeck, "Context models for palette images," in *Proc. IEEE Data Compression Conf.*, J. Storer and M. Cohn, Eds. Snowbird, UT, 1998, pp. 309–318.
- [50] J. Ostermann, E. S. Jang, J.-S. Shin, and T. Chen, "Coding of arbitrarily shaped video objects in MPEG-4," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Santa Barbara, CA, 1997, pp. 496–499.
- [51] D. Saupe and H. Hartenstein, "Lossless acceleration of fractal image compression by fast convolution," in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Sept. 1996.
- [52] J. Kominek, "Waterloo BragZone," Available [Online] <http://links.uwaterloo.ca/bragzone.base.html>.
- [53] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 4, 1989.
- [54] J.-M. Morel and S. Solimini, *Variational Methods in Image Segmentation*. Boston, MA: Birkhäuser, 1995.
- [55] D. Saupe, "Fractal image compression via nearest neighbor search," in *Conf. Proc. NATO ASI Fractal Image Encoding and Analysis*, 1998, pp. 95–114.



Dr. Hartenstein received the Capocelli Award from the IEEE Data Compression Conference 1997 for the paper (with M. Ruhl) "Optimal fractal coding is NP-hard."



Mr. Ruhl received the Capocelli Award (with H. Hartenstein) from the IEEE Data Compression Conference 1997.

Hannes Hartenstein (S'97–A'99) received the Diploma degree in mathematics in 1995 and the Ph.D. degree in computer science in 1998, both from Albert–Ludwigs Universität, Freiburg, Germany.

He was Erasmus Scholar with the University of East Anglia, Norwich, U.K., in 1991–1992. He is now a Research Staff Member with the Computer & Communications Research Laboratories, NEC Europe, Heidelberg, Germany. His research interests include image/video processing, mobile communications, and theoretical computer science.

Matthias Ruhl received the M.S. degrees in mathematics and computer science from the Albert–Ludwigs-Universität Freiburg, Germany, in 1997 and 1998, respectively. He is currently a graduate student in computer science at the Massachusetts Institute of Technology, Cambridge.

He worked on theoretical and practical aspects of fractal image compression, and is also interested in algorithms, complexity theory, cryptography and logic.

Dietmar Saupe received the Dr. rer. nat. degree in 1982 and the Habilitation degree in 1993, both from the University of Bremen, Bremen, Germany.

He was Visiting Assistant Professor of mathematics at the University of California, Santa Cruz, from 1985 to 1987, Assistant Professor at the University of Bremen from 1987 to 1993, and Professor of computer science at the Albert–Ludwigs University, Freiburg, Germany, from 1993 to 1998. He has been a Professor at the University of Leipzig, Leipzig, Germany, since 1998. His research interests are in image processing, computer graphics, visualization, experimental mathematics, and dynamical systems. He is author and editor of several books on chaos and fractals.

Dr. Saupe is a Member of the IEEE Signal Processing Society and ACM SIGGRAPH.